

# CLI Publishing Options File

If needed, you can specify multiple file or table targets as part of a single CLI job. In your CLI command, the path on the Trifacta node to this JSON file is specified as the `publish_opt_file` parameter, as in the following:

```
./trifacta_cli.py run_job --user_name <trifacta_user> --password <trifacta_password> --job_type spark --data redshift-test/datasources.tsv --script redshift-test/script.cli --cli_output_path ./job_info.out --profiler on --publish_opt_file /json/publish/file/publishopts.json
```

The file `publishopts.json` contains the specification of the targets.

**Tip:** To specify this file, you can run this job through the application. After the job has completed, download the CLI script from the Recipe panel in the Transformer page. The downloaded `publishopts.json` file contains the specification for the targets you just executed. See *Recipe Panel*.

Example `publishopts.json` file:

```
{
  "file": [
    {
      "path": "hdfs://hadoop:50070/trifacta/queryResults/admin@trifacta.local/POS-r01.csv",
      "action": "create",
      "format": "csv",
      "header": true,
      "asSingleFile": true,
      "compression": "none"
    },
    {
      "path": "hdfs://hadoop:50070/trifacta/queryResults/admin@trifacta.local/POS-r01.json",
      "action": "create",
      "format": "json",
      "header": false,
      "asSingleFile": false,
      "compression": "none"
    }
  ],
  "hive": [
    {
      "databaseName": "default",
      "tableName": "POS-r01",
      "action": "overwrite"
    }
  ]
}
```

**NOTE:** All of the following properties require valid values, unless noted.

## File targets:

Property	Description
<code>path</code>	Full path to the target file. Path must include the protocol identifier, such as <code>hdfs://</code> and the port number.

action	<p>The action to take on the file. Supported actions:</p> <ul style="list-style-type: none"> <li>• <code>create</code> - Create a new file with each subsequent publication. Filenames for subsequent job runs are appended with the job number identifier.</li> <li>• <code>append</code> - The results of each subsequent job run are appended to the existing file contents.</li> <li>• <code>replace</code> - The results of each subsequent job run replace the same file. Previous job run results are lost unless moved out of the location.</li> </ul> <p>Some limitations apply to these options. See <i>Run Job Page</i>.</p>
format	<p>Output format for the file. Supported formats:</p> <ul style="list-style-type: none"> <li>• <code>csv</code></li> <li>• <code>json</code></li> <li>• <code>avro</code></li> <li>• <code>pqt</code></li> </ul>
header	<p>If set to <code>true</code>, then output files in CSV format include a header row. Headers cannot be applied when compression is enabled.</p>
asSingleFile	<p>If set to <code>true</code>, then output files are written to a single file.</p> <p>If set to <code>false</code>, then the output files are written to multiple files as needed.</p>
compression	<p>(optional) This property can be used to specify any compression to apply to a text-based file. Supported compression formats:</p> <ul style="list-style-type: none"> <li>• <code>gzip</code></li> <li>• <code>bzip2</code></li> <li>• <code>snappy</code></li> </ul> <p>If this is not specified, then no compression is applied to the output file.</p>

### Hive targets:

Property	Description
databaseName	<p>Name of the database.</p> <div style="border: 1px solid #ccc; padding: 5px; margin-top: 10px;"> <p><b>NOTE:</b> The database must contain at least one table.</p> </div>
tableName	<p>Name of the table in the database to which to write.</p>
action	<p>The write action to apply to the table. Supported actions:</p> <ul style="list-style-type: none"> <li>• <code>create</code> - Create a new table with each subsequent publication. Table names for subsequent job runs are appended with a timestamp.</li> <li>• <code>append</code> - The results of each subsequent job run are appended to the existing table contents.</li> <li>• <code>replace</code> - The results of each subsequent job run are written to the same table, which has been emptied. Previous job run results are lost unless moved out of the location (<code>dropAndLoad</code>).</li> <li>• <code>overwrite</code> - The results of each subsequent job run are written to a newly created table with the same name as the output table from the previous job run (<code>truncateAndLoad</code>).</li> </ul> <p>Some limitations apply to these options. See <i>Run Job Page</i>.</p>